XIX IMEKO World Congress
Fundamental and Applied Metrology
September 6−11, 2009, Lisbon, Portugal

DESIGN OF THE PROTOTYPE OF PLD AUTO TEST PLATFORM

Senzu Shen, Hua Li, Zhengle Shi, Minghu Zhang and Qian Liu

Wuhan Digital Engineering Institute, Wuhan, P. R. China, cmmt2008@gmail.com

Abstract − There're mainly two different situations for testing PLD (Programmable Logic Device) devices. One is testing of blank chips that haven't been programmed; the other is testing of ASIC (Application Specific Integrated Circuit) devices that have been programmed. This paper presents a test method for programmed ASIC devices, including auto test and ATPG (Auto Test Pattern Generation) method for both known and unknown logics of programmed ASIC devices.

Keywords: PLD, auto test, ATPG

1. INTRODUCTION

Microelectronic measurement and test plays an important role in product quality control. There're mainly two different situations for testing PLD (Programmable Logic Device) devices. One is testing of blank chips that haven't been programmed; the other is testing of ASIC (Application Specific Integrated Circuit) devices that have been programmed. These two situations are quite different in test purpose and technical method. The former focuses on the availability of programmable resources that belongs to raw materials test, while the latter aims at the reliability of functions and parameters of the programmed devices that belongs to product quality test. As a result, the latter is much more important than the former one.

This paper presents an auto test method for the latter situation. The presented method only takes no more than 30 minutes to develop a test program while traditional methods should at least take 3-5 days. This greatly reduces the development cycle of a test program.

2. DESIGN OBJECTIVE

ASICATPGG auto test operation platform is an auto test system which combines PC and S10 test system together and is supported by several application software, specially applied for ASIC devices. It can generate S10 test program automatically according to logic description, program files or known golden devices. The generated test program can be loaded into S10 test system automatically.

Specifications of the test platform are listed in table 1.

Table 1. Specifications of the presented test platform.

Functional test coverage	100%
Program generation time	Less than 30 minutes
AC error (depends on test system)	±2ns for S10
DC error (depends on test system)	0.2%-0.5% for S10

3. HARDWARE PROTOTYPE

The hardware is composed of a PC, S10 LSI (Large Scale Integrated circuit) test system and a simulator as shown in fig. 1.

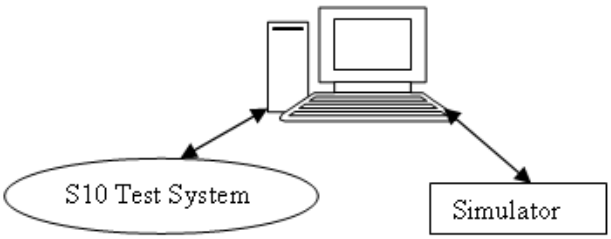

Fig. 1 Hardware composing

4. SOFTWARE PROTOTYPE

Operating system: DOS (PC) and M^3 (S10)

Program design language: ABEL, Turbo C (PC) and FACTOR (S10)

Communication between DOS and M^3 is realized by heterogeneous machine communication program.

5. COMMUNICATION RULES

BNF (Backus-Naur Form) is used, assume:

::= stands for "be defined as";

Characters in single quotation marks are final definition, such as characters and commands;

Characters in sharp-angled brackets are optional terms;

Characters in {} stands for repeated terms (0 or multi times);

"|" stands for optional;

Suffix :n stands for repeated times;

ASCIIxx hex means that "xx" is a hexadecimal ASCII value;

<binary-digit>::='0' | '1'

<octal-digit>::='0' | '1' | '2' | '3' | '4' | '5' | '6' | '7'

<digit>::=<octal-digit> | '8' | '9'

<hex-digit>::= <digit> | 'A' | 'B' | 'C' | 'D' | 'E' | 'F'

<Full>::=<ASCII 20 hex>

<chara>::='A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' |

'J' | 'K' | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' |

ISBN 978-963-88410-0-1

'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z '

In addition, interactive operation command grammar, communication and transfer grammar, file transfer flow grammar and character set transfer method are depicted as follows.

5.1 Grammar of Interactive Operation Command

Cs10 stands for all the system level commands of S10, Cpc stands for all the system level commands for PC, Tpc stands for commands for switching PC to S10 method, Rpc stands for commands for switching S10 method to PC. Consequently, the grammar of interactive operation command will be:

<format>::=<Cs10> | <Rpc> | <Tpc> | <Cpc>

5.2 Communication Transfer Protocol

The communication transfer protocol is RS-232 standard.

5.3 Communication and Transfer Grammar

<Linesetting>::=<band>','<parity>','<Data-bits>','<stop-bits>',' <c-port>

<band>::=<digit> [<digit>:4]

<parity>::='N' | 'O'

<Data-bits>::=<digit>

<stop-bits>::=<digit>

<c-port>::=<chara> [[<chara> | <digit>]:3]

5.4 Grammar of Communication Subcommand

<TFTP>::=<TFTpc> | <TFTPs10>

<TFTpc>::='PGdn' | 'PGup'<full> '2'<full> <file-spec>

<file-spec>::=[<path-name>]<file-name> '.' <type>

<path-name>::=[<nodename>]<device>

<nodename>::=<chara>[[<chara> | <digit>]:5]

<device>::=<chara>[[<chara> | <digit>]:5]

<file-name>::=<chara>[[<chara> | <digit>]:5]

<type>::=<chara> | <digit>[[<chara> | <digit>]:2]

<TFTPs10>::='File'<Full>'send' | 'Rec'<Full>

[<pathname>]<Full><filename><Full><type>

Where <TFTPs10> is only used in S10 operation mode, <TFTpc> is only used in PC mode.

5.5 Grammar of File Transfer Flow

<stream>::=<begin><file-name><length>{<contents>}<end><check-sum>

<begin>::=<ASCII 02 hex>

<length>::=<digit>[<digit>:7]

<contents>::=<binary-digit>

<end>::=<ASCII 03 hex>

<check-sum>::=<digit>[<digit>:7]

5.6 Transfer Method of Character Set

Since S10 character set is the subset of PC character set, character set conversion is accomplished by S10 communication program.

Rule No. 1: All small letters are converted to capital letters (subtract 20hex);

Rule No. 2: Characters that are not defined in S10 are converted to blanks.

6. SYSTEM WORKING FLOW AND DATA FLOW

The presented system is a complicated system which includes heterogeneous machine communication, multi-language compiling and two operating systems. The application software can be divided into two parts: one runs in DOS system of a PC, the other runs in M^3 system of S10.

Modules run in DOS include input generation, test pattern generation and S10 test program generation.

Modules run in M^3 include auto compiling, loading and test module.

Working program (entrance): ASICATPGG

The working flow of the presented system is shown in fig. 2.

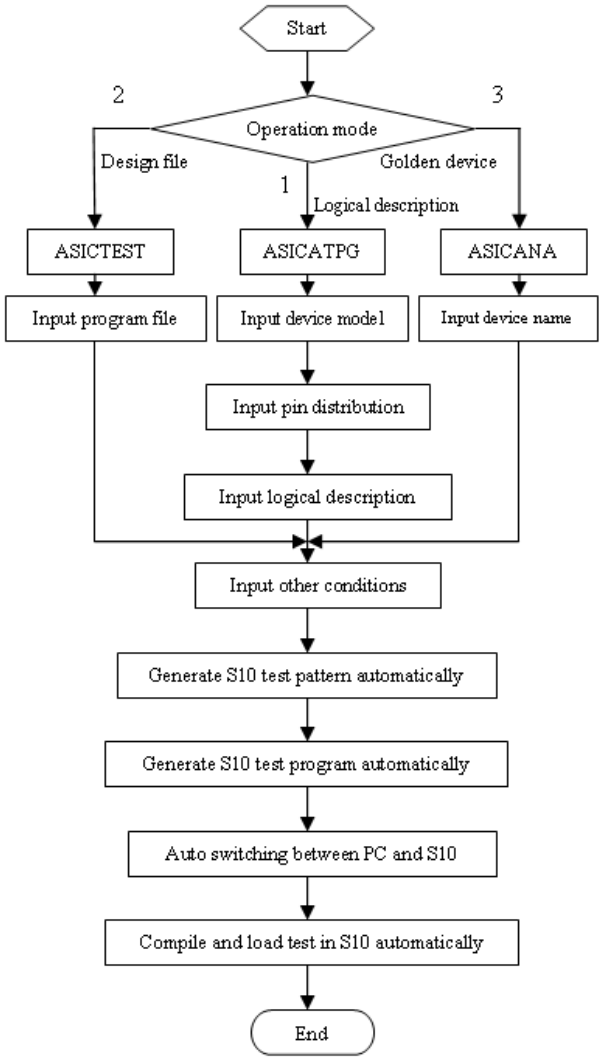

Fig. 2 System working flow

The data flow is depicted in fig. 3.

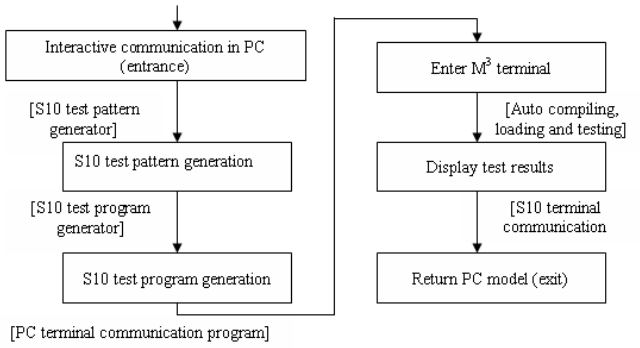

Fig. 3 Data flow

Application of ASICATPGG needs to satisfy at least one of the following conditions: a known type golden device/ program files/ known logic expressions of pins.

The golden device is a known device in good condition. Program files are logical description files burning through programmer. Logical expressions are logical descriptions of device pins. The key point of auto generation test pattern is that they are all inputs of logical simulation.

7. REALIZATION

7.1 Auto Generation of Test Program

ASIC test program auto generation is the key point of this project. Test program auto generation can be divided into test pattern auto generation and test program auto generation.

7.1.1 Test Pattern Auto Generation

Test pattern auto generation relies on the following factors: test pattern generator which simulates the golden device/ program files/ known logic expressions of pins; auto generation pattern; test pattern generation; sequencing circuit; state enumeration; logic circuit and enumeration.

The principles of state enumeration are listed below:

Declaration No. 1: In functional simulation of any IC devices there always are some pins that have fixed values; also there are some other pins whose values require enumeration and can be optimized.

Declaration No. 2: In functional simulation of any IC devices, any useful pins must belong to one and only one of the above situations.

7.1.2 Generation Arithmetic

Assume I is the vector set of analog input, T_V is the vector set (input/output) obtained after simulation; I_Q stands for the vector subset composed of Q pins in I set. I_S stands for the enumeration vector set composed of S pins. I_U stands for the optimization vector set composed of U pins in I set. f stands for functional simulation of devices. O_o stands for the simulation output result using I_i as the input excitation signal. C stands for a comparison operation. G stays as before. Then the arithmetic model of test vector would be:

$I=I_Q+I_S+I_U=\{I_i\}$;

$G(I_i)=f(I_i),C(O_i,O_o)$;

$T_V=\{G(I_i)\}=\{T_{Vj}\}$;

among which i<=j, i=1,2,3,…(s2*2u) , j=1,2,3,…(s2*2u)+T;

T stands for the total sum of state extension for i vectors.

7.1.3 Calculation formula of numerical value

Assume a DUT (Device Under Test) has n input pins, where the mth bit belongs to fixed index, the rth bit belongs to enumeration index and the pth bit belongs to optimization index. In addition, the optimization method is 0 step, 1 step, then:

$$Ii=\begin{cases}\sum_{j=1}^{m}2n-j\times aj+i & when\ p=\\ \sum_{j=1}^{m}2n-j\times aj+k\times 2p+2i\%p & for\\ \sum_{j=1}^{m}2n-j\times aj+k\times 2p+\sum_{j=0}^{p-1}2j-2i\%p & \end{cases} \quad (1)$$

among which n=m+r+p； i=0,1,2,…,2r-1,2r,…,2p×2r； j=1,2,3,…,m ； k=0,1,2,…,2r-1 ； % stands for a modulo operation; a_j is a Boolean function:

$$a_j=\begin{cases}0 & when\ the\ fixed\ value\ is\ active\ low\ o\\ 1 & when\ the\ fixed\ value\ is\ active\ high\ o\end{cases} \quad (2)$$

"Step 0" pattern is defined as below:

1…110
1…101
…
…
…
01…11

"Step 1" pattern is defined as below:

0…001
0…010
…
…
…
10…00

7.1.4 Auto Generation of Test Program

Test program includes at least functional test, AC parameter test such as TPDLH and TPDHL and DC parameters test such as ICC, ICCL, ICCH, IOS, IIL, IIH, II, VIK, VIL, VIH, VOL and VOH. As the testability design technique is used in auto generation of test program, so all the parameters can be described quantificationally.

7.2 Relative Key Technique

7.2.1 Multi-use of sequencing circuit pattern

The state of a sequencing circuit is uncertain after power up. As a result there is an initialization problem. Any set of test pattern of sequencing circuit may cause different results under different conditions. Consequently there is a technical problem for state reset. The state reset and initialization are both the same question whose final purpose is to make sure that the sequencing circuit come to a certain state. Usually there're two methods for obtaining the initialization vectors of the sequencing circuit: dynamic force setup and interactive booting. The dynamic force setup is proper for GAL (Generic Array Logic) devices while interactive booting is proper for PAL (Programmable Array Logic) devices.

With regard to multi-use of sequencing circuit pattern, the initialization vectors of sequencing circuit are induced. Dynamic force setup and interactive booting can either be used to obtain the initialization vectors of the sequencing circuit.

7.2.2 Auto position technique in DC test

Despite of the relative language requirement of test vectors and test program, rational and accurate arrangement of such tests is also an important question. The most difficult part is the uncertainty of the output parameter test. Because during auto test program design, we don't know which pin is being tested and in which vector it should be high or low. A usual way is to use the auto position technique. The test program will search the needed state automatically and run the test. There's no doubt that this strategy is totally reasonable. As long as the truth table is complete, the auto position technique can be widely used for generic devices.

In ASIC auto test of the presented platform, the uncertainty of DC parameter test is also a problem. Similarly, the auto position technique is adopted.

7.2.3 Auto switching between two systems

Auto switching between DOS and M^3 is accomplished by heterogeneous machine communication program. The communication transfer protocol is:

- RS-232 standard signal;
- Character format
- Baud rate: 9600, verification method: N, data bit: 8, stop bit: 1.

The 8 bit data bit is significant because it makes S10 possible to transfer 1 byte by three times.

7.2.4 Testing for devices with unknown logics

Devices with unknown logics have two types. One is devices with unknown logics; the other is devices with unreadable logics. A method of reading burning logic node pattern is adapted combine with the internal information of the PLD device; finally the burning logic description can be analyzed based on logic simulation and logic synthesis. The analyzed logic description is only used in test generation other than other purposes. The logic description file will disappear after test generation for the safety of customer property.

8. OTHER INSTRUCTIONS

8.1 Optimization Rules

The optimization function provided by ASICATPGG aims at improving test pattern quality and functional test coverage and reducing the length of test pattern. Usually ASIC pins are optimized by group, such as data pins group, control pins group. Generally speaking, all data pins should be optimized while other group depends.

8.2 Applications of the Platform Prototype

PLD auto test platform prototype is especially designed for S10 large scale integrated circuit test system, PC 486 and DMU-2000 simulator. Particularly, any universal test system can be used to replace S10 and any types of popular PC can be used to replace PC486 as long as the simulator is the same. DMU-2000 can be used in all kinds of PC. When other test system and PC are chosen, the PC related application software is compatible, only the test system related software needs to change accordingly, such as the system operation commands and test languages. Actually these changes are not tough because of the presented prototype. The corresponding equivalent sentences can be used.

9. CONCLUSIONS

The technical difficulties of the presented prototype are test pattern generation and auto test pattern generation (ATPG).

No public reports/papers relative with the presented prototype have been published yet. The innovative points of the presented platform design are listed below:

a. The "state enumeration" of test pattern generation arithmetic is unique.

b. The multi-use of sequencing circuit pattern and auto position of DC test is simple and reasonable.

c. Devices with unknown logics can also be tested.

Finally the prototype of PLD auto test platform is completely realized and the presented method has been successfully applied in over 10 ASIC devices including 7022 etc. The prototype can be used in formal product design.

REFERENCES

[1] Zainalabedin Narabi: "*VHDL： Analysis and Modeling of Digital Systems*", McGraw-Hill Professional, United States of America, 1997.

[2] Miron Abramovici, Melvin A. Brener, Arthur D. Friedman: "*Digital Systems Testing and Testable Design*", Wiley-IEEE Press, New Jersey, 1994.

[3] Jean Michel Berge, Alain Fonkna, Serge Maginot, Jacques Roulliard: "*VHDL Designer's Reference*", Springer, 1992.